Key-Card Door Lock

Written By: Charles Platt

TOOLS:

- Multimeter and test leads (1)
- Soldering iron and solder (1)
- Wire cutters and strippers (1)

PARTS:

- Phototransistors, visible spectrum (6)
  Everlight part #PT334- 6C or RadioShack #55053303
- LEDs, white, water-clear (3)
  typical 3.3V forward voltage, minimum, 5,000mcd with 30° dispersion; Kingbright #WP7104VW1C or RadioShack #55050632
- Slide switches, subminiature DPDT (6)
  C&K #OS202013 MT5QN1
- Toggle switch, DPDT, centeroff momentary, minimum 3A (1)
  Mountain Switch #108-0005-EVX
- Resistor arrays, bussed, 5.6K (2)
  Xicon #265-5.6K-RC
- Logic chip, triple 3-input AND gates, type 74HC11 (1)
  Texas Instruments #SN74HC11N
- Darlington transistor, 8A, type 2N6043 (1)
  On Semiconductor #2N6043G
- Mini lever snap switches, SPDT, 5A (4)
  maximum 20g operating force, Mountain Switch #101-1206-EV
- Diodes, general purpose, 3A (2)
  Vishay #SRP300G-E3/54
- Voltage regulator, 5V, type LM7805 (1)
  Fairchild Semiconductor #LM7805CT or RadioShack #276-1770 » Capacitors.
- Capacitors, multilayer ceramic (1)
  For smoothing the 5V output
- Gearmotor, 9V or 12V DC, minimum 10rpm (1)

10rpm (1)
ServoCity #RZ12-300-10RPM or check eBay for substitutes

- 9V battery (1)

- Breadboard (1)
Optional, recommended for testing

- Jumper wires (1)
Optional, recommended for testing

SUMMARY

Almost anyone can learn to pick a typical door lock. The most common method is to jiggle the tumblers by inserting a tool known as a "rake," while applying turning force with a tension wrench. When the tumblers are aligned, the cylinder rotates and you can walk right in.

When I first started thinking about a pick-proof electronic lock, I imagined a totally automated system using a key card and a motor that would pull a deadbolt aside. However, a motor that was powerful enough to overcome the likely friction would require either 115V AC (vulnerable to power outages) or a substantial battery.

It would be cheaper and simpler to use an off-the-shelf deadbolt that's moved manually by turning an external knob. I could block or unblock the movement of the deadbolt using a small, low-power DC gearmotor to turn a lever (see Step 1).

For the key card, I imagined a plastic rectangle punched with a unique pattern of holes. The card would slide into a slot with a light source to shine through the holes to phototransistors beneath. Additional electronics would decide whether the pattern was correct. If I needed to rekey the lock, I could make a new card with a different pattern, and reprogram the electronics to match.

The word "reprogram" suggested I'd need a microcontroller to complete this project. Or would I? Maybe a set of switches could do the job more simply and cheaply. The first step toward figuring this out was to review the way in which phototransistors generally behave.

Step 1 — Key-Card Door Lock
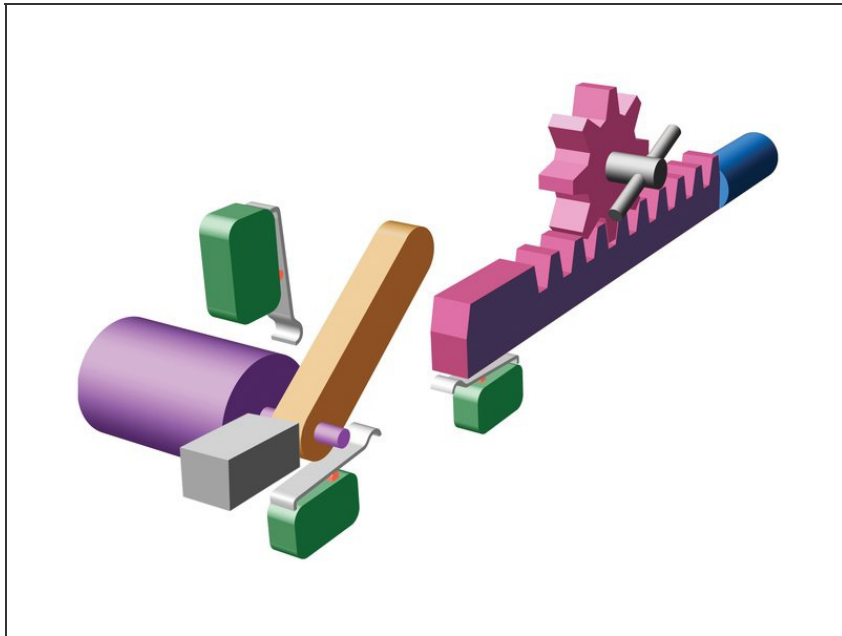
- In this simplified rendering, a deadbolt (blue) is turned by an external knob (gray). The sliding shaft (red) is either blocked or freed by a lever (yellow) driven by a motor (purple) mounted inside the door.
- Limit switches (green) stop the motor when the lever reaches either end of its travel.
- A third switch beneath the sliding shaft shuts down power when the deadbolt retracts, to prevent the motor from fouling it and burning out.
- A fixed block (gray) resists the force transmitted through the lever if someone tries to jimmy the closed lock.

Step 2 — Phototransistors Basics
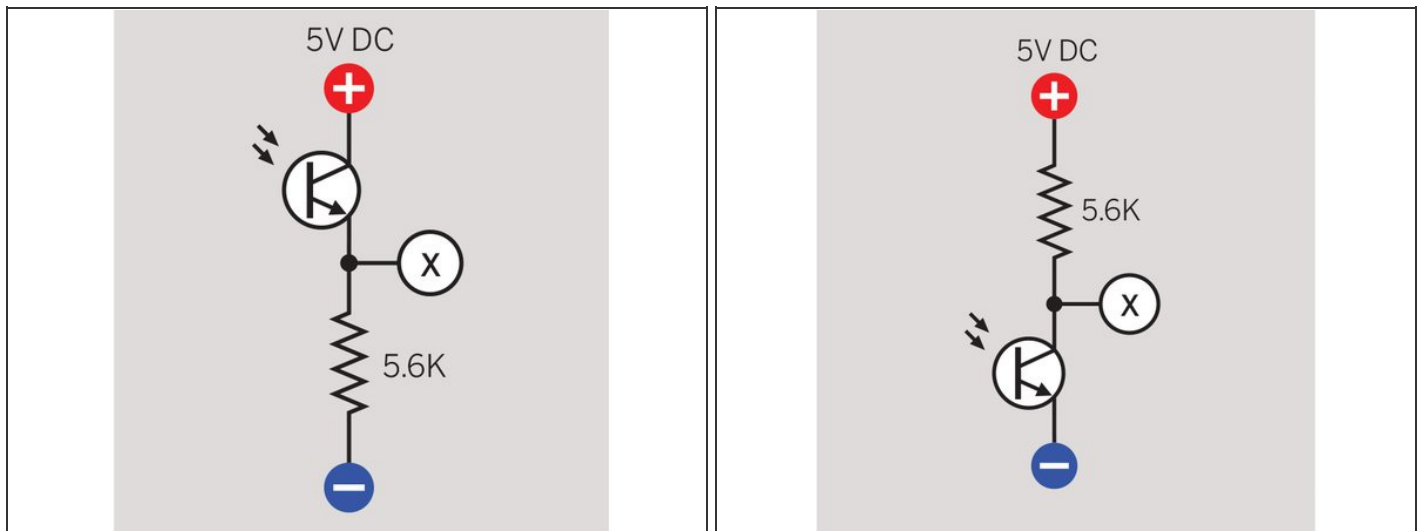
- In the first image, the phototransistor is the circle with arrows suggesting incoming light. When the light is off, the phototransistor allows only a tiny leakage current between emitter and collector, and the voltage between point X and ground is almost zero.

- As the light increases, the phototransistor junction becomes saturated, just like an NPN transistor when the base is highly biased. The effective internal resistance drops, and the voltage at point X rises close to the supply voltage (assuming any electronics attached to X have high impedance).

- I found a 5.6K resistor worked well with the phototransistor I chose for this project.

- If the positions of the phototransistor and the resistor are reversed, the outcome is too. In the second image, when the phototransistor is dark and blocks current, the voltage at X is high. When light shines on the phototransistor, the voltage at X goes low.

- So, depending on wiring, a phototransistor can give a high output either in the light (first image) or in the dark (second image). Thus, if phototransistors are wired as in image 1 to match the holes in a key card, and in image 2 to match the blocked areas of the card, all their outputs will go high when the card's pattern is correct.

Step 3
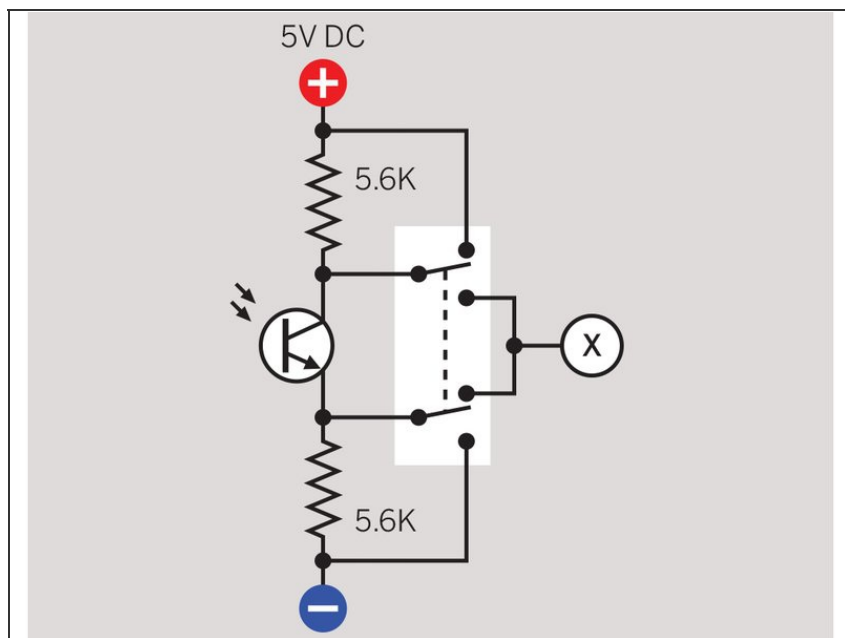
- This can be verified by ANDing all the phototransistor states via a logic chip, because an AND gate only outputs high if all its inputs are high.

- A hardwired array of phototransistors is not reprogrammable, but I saw that a DPDT switch could solve the problem. In its "up" position the switch emulates the circuit in Step 2's first image. In its "down" position the switch emulates the circuit in Step 2's second image.

- If the key card pattern is changed, the switch positions are changed to match.

- Phototransistors and subminiature slide switches are cheap, so the electronics for this project shouldn't cost more than $20. The motor and deadbolt are extra, but I found small DC motors on eBay for $5 or $6 each.

Step 4 — Combinations
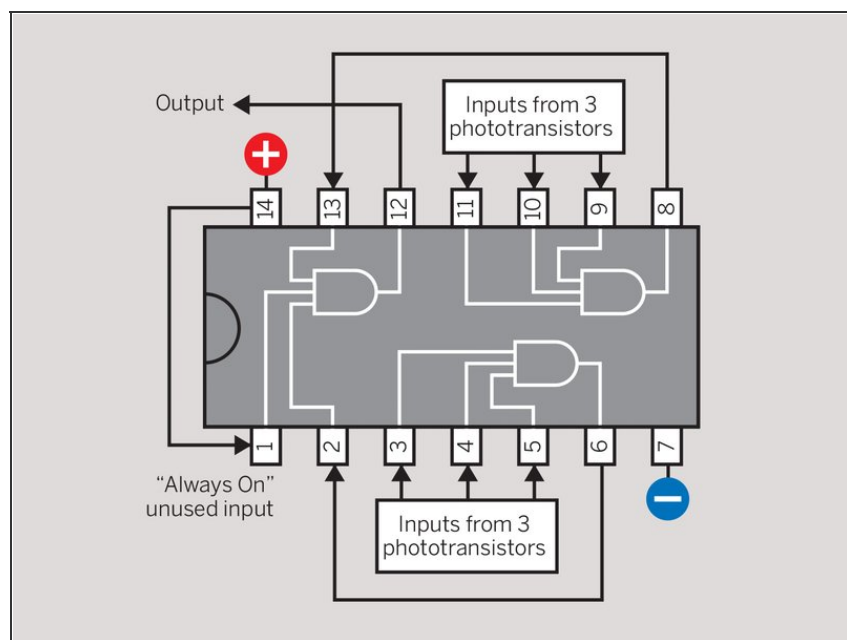
- Now to decide how many phototransistors to use. Sixteen in a 4×4 grid would allow 65,536 possible combinations, which sounded reasonably secure. But 16 phototransistors, 16 switches, and 32 resistors is a lot to wire. Then I realized most of the holes in the card could be dummies.

- Six phototransistors would still provide 64 possible combinations for the lock. I would be the only person who knew which holes were dummies, and which were not.

- Next, how to AND 6 phototransistor outputs? This is easy using a 74HC11 chip with three 3-input AND gates. The image shows how they're chained to give a single output.

- These phototransistors can be connected directly to the chip because they comply with its requirement for a high input no less than 3.5V and a low input no greater than 1V DC. Use a meter to verify your phototransistors also meet this requirement.

Step 5 — Additional Switches
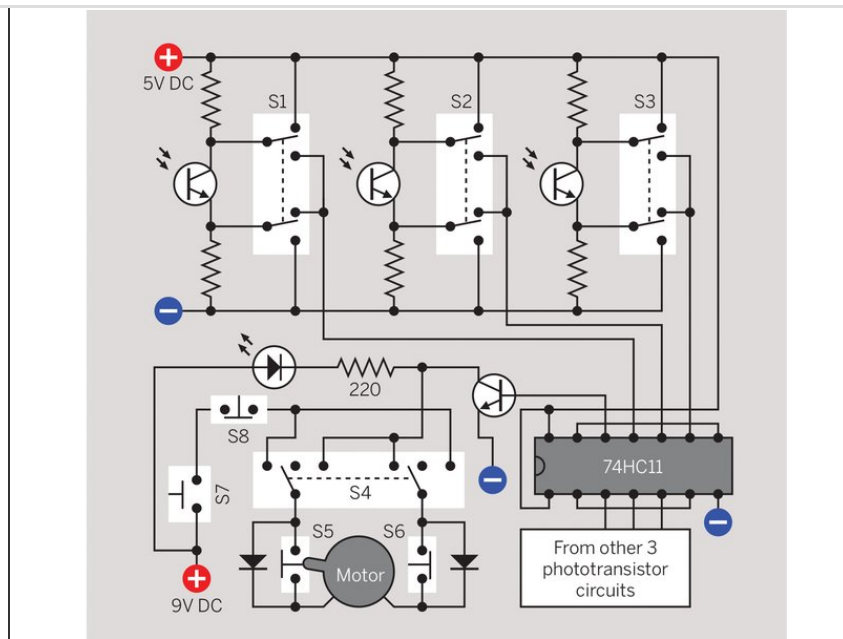
- The chip's output controls a Darlington transistor, which powers the motor (image). The motor stops itself at each end of its travel by opening one of a pair of limit switches, marked S5 and S6. (If you're unclear about how limit switches work, you'll find a detailed explanation in my book *Make: Electronics*.)

- S4 is a manual DPDT switch, spring-loaded to return to its center-off position. It starts the motor in one direction or the other by bypassing either S5 or S6 through a rectifier diode. If your motor doesn't respond appropriately, just swap the wires to its inputs.

- The purpose of S7 is to minimize battery drain by keeping the circuit powered down until a key card is inserted. This switch should be positioned so the card closes it when slid fully in.

- Because the motor will burn out if it tries to turn the lever to block the deadbolt when the deadbolt has already been retracted, I needed one more fail-safe switch. This is S8, which shuts off the power when the opening deadbolt rubs against it.

Step 6 — More Details
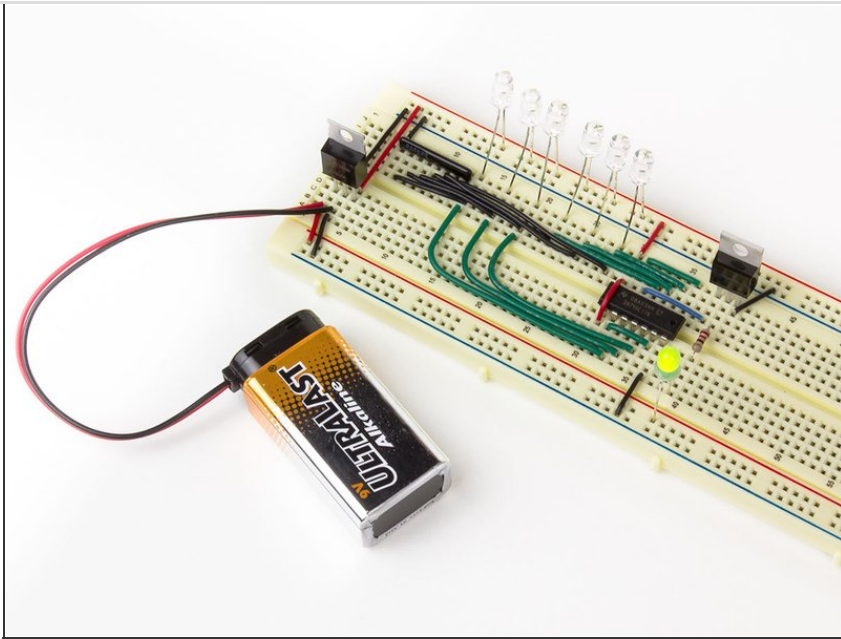
- If you use a really small gearmotor rated for 9 or 12V DC, the whole lock can be powered by a 9V battery. An LM7805 voltage regulator will reduce its output to the 5V required by the phototransistors and the chip. The LM7805 datasheet, online, will show you how it should be used in conjunction with a couple of bypass capacitors.

- Because the motor is reversed by changing the polarity of the power, I couldn't add a protection diode to divert the surge that the motor takes when it starts or stops.

- If you find that the power spike disturbs your logic chip, try putting a 0.1 microfarad (F) capacitor between the positive power supply to the chip, on pin 14, and the negative supply on pin 7.

- The resistors in series with the phototransistors are all 5.6K. To simplify wiring and save space, I used a pair of resistor arrays. Check the datasheet online for the resistor array to see how the resistors are wired inside.

- You may want to add an LED indicator outside the lock, tied to the output from the Darlington transistor, to let you know that the key has been accepted and the position of the deadbolt can now be

changed.

- Last but not least, you'll need a light source to activate the phototransistors.

- Many phototransistors are only sensitive to infrared, but the type I chose will respond to a very broad spectrum, so you should be able to use 3 bright white LEDs to shine down upon the key card. Wire them in series, and you can power them with 9V DC and no additional load resistor.

Step 7

- As for interfacing the motor with the deadbolt, this mechanical issue is outside the scope of this column, but the setup should look basically like the image in Step 1. I leave it to you to determine the details of fabricating a card slot and mounting it in the door.

- Can this lock be broken? Of course! If someone uses enough brute force, almost any door will yield. Still, the electronic lock will be considerably more secure than a traditional lock containing mechanical tumblers.

This project first appeared in [MAKE Volume 33](#), page 92.

This document was last generated on 2013-01-30 10:47:58 PM.